# Assignment 4

## Student Name: Diksha Pandey UID:22BCS10359

## Branch: BE-CSE (General) Section/Group: FL\_IOT-602-A

## Semester:6th Date of Performance: 21-02-25

## Subject Name: Advanced Programming Lab-2 Subject Code: 22CSP-351

# **Aim:** [1763. Longest Nice Substring](https://leetcode.com/problems/longest-nice-substring/)

# Implementation/ Code:

# class Solution {

# public:

# string longestNiceSubstring(string s) {

# string output = "";

# int count = 0;

# for(int i = 0;i<s.length();i++){

# int smallMask=0;

# int largeMask = 0;

# char ch = s[i];

# int chint = 0;

# if(ch>=65 && ch<=90){

# chint = ch-'A';

# largeMask = 1<<chint;

# }

# else{

# chint = ch-'a';

# smallMask = 1<<chint;

# }

# for(int j = i+1;j<s.length();j++){

# ch = s[j];

# if(ch>=65 && ch<=90){

# chint = ch-'A';

# largeMask |= 1<<chint;

# }

# else{

# chint = ch-'a';

# smallMask |= 1<<chint;

# }

# 

# if((smallMask^largeMask) == 0){

# if(count<j-i+1){

# count = j-i+1;

# string temp(s.begin()+i,s.begin()+j+1);

# output = temp;

# }

# }

# }

# }

# return output;

# }

# };

# Screenshot 2025-02-21 at 9.53.32 AM.png

# Output:

# **Aim:** [190. Reverse Bits](https://leetcode.com/problems/reverse-bits/)

# Implementation/ Code:

# class Solution {

# public:

# uint32\_t reverseBits(uint32\_t n) {

# 

# n = ((n & 0xffff0000) >> 16) | ((n & 0x0000ffff) << 16);

# n = ((n & 0xff00ff00) >> 8) | ((n & 0x00ff00ff) << 8);

# n = ((n & 0xf0f0f0f0) >> 4) | ((n & 0x0f0f0f0f) << 4);

# n = ((n & 0xcccccccc) >> 2) | ((n & 0x33333333) << 2);

# n = ((n & 0xaaaaaaaa) >> 1) | ((n & 0x55555555) << 1);

# 

# return n;

# 

# }

# };

# Output:

# Screenshot 2025-02-21 at 9.56.55 AM.png

# **Aim:** [191. Number of 1 Bits](https://leetcode.com/problems/number-of-1-bits/)

# Implementation/ Code:

# class Solution {

# public:

# int hammingWeight(uint32\_t n) {

# int res = 0;

# for (int i = 0; i < 32; i++) {

# if ((n >> i) & 1) {

# res += 1;

# }

# }

# return res;

# }

# };

# Output:Screenshot 2025-02-21 at 10.00.04 AM.png

# **Aim:** [53. Maximum Subarray](https://leetcode.com/problems/maximum-subarray/)

# Implementation/ Code:

# class Solution {

# public:

# int maxSubArray(vector<int>& nums) {

# int res = nums[0];

# int total = 0;

# for (int n : nums) {

# if (total < 0) {

# total = 0;

# }

# total += n;

# res = max(res, total);

# }

# return res;

# }

# };

# Output:Screenshot 2025-02-14 at 2.09.48 PM.pngScreenshot 2025-02-21 at 10.02.13 AM.png

# **Aim:**[240. Search a 2D Matrix II](https://leetcode.com/problems/search-a-2d-matrix-ii/)

# Implementation/ Code:

# class Solution {

# public:

# bool searchMatrix(vector<vector<int>>& matrix, int target) {

# int n=matrix.size();

# int m=matrix[0].size();

# int row=0,col=m-1;

# while(row<n && col>=0){

# if(matrix[row][col]==target){

# return true;

# }else if(matrix[row][col]<target){

# row++;

# }else{

# col--;

# }

# }

# return false;

# }

# };

# Output:Screenshot 2025-02-14 at 2.12.28 PM.pngScreenshot 2025-02-21 at 10.04.13 AM.png

# **Aim:** [372. Super Pow](https://leetcode.com/problems/super-pow/)

# Implementation/ Code:

# class Solution {

# private:

# int solve(int base, int power, int mod) {

# int ans = 1;

# while (power > 0) {

# if (power & 1) {

# ans = (ans \* base) % mod;

# }

# base = (base \* base) % mod;

# power >>= 1;

# }

# return ans;

# }

# public:

# int superPow(int a, vector<int>& b) {

# a%=1337;

# int n = b.size();

# int m = 1140;

# int expi = 0;

# for(int i : b){

# expi = (expi\*10+i)%m;

# }

# if (expi == 0) {

# expi = m;

# }

# return solve(a,expi,1337);

# }

# };

# Output:Screenshot 2025-02-21 at 10.06.16 AM.png

# 

# 